

CS3 Rubric

Case Study Rubric <ul style="list-style-type: none">• Due: TBD• Submission format: upload pdf and link to GitHub repo to Canvas
General Description: <ul style="list-style-type: none">• Submit to Canvas your PDF and a link to your GitHub repository.
Why am I doing this? <ul style="list-style-type: none">• This study is an opportunity to build upon your understanding of machine learning by developing your own deep learning model using PyTorch.• It mirrors real-world tasks you may face in both academic research and applied AI roles, especially in medical imaging.
What am I going to do? <ul style="list-style-type: none">• With foundational knowledge of pretrained PyTorch models already covered, you are now tasked with training your own model from scratch that segments kidney tumors from a provided dataset.• You must ensure your model avoids overfitting via proper use of a training-test split.• Your submission will include:<ul style="list-style-type: none">- Written Portion PDF- GitHub Repository with code and artifacts
How will I know I have succeeded? <ul style="list-style-type: none">• You will meet expectations on this case study when you fulfill the criteria outlined in the rubric below.
Formatting <ul style="list-style-type: none">• Written Portion: Submit the written portion as a PDF document.• Data & Code:<ul style="list-style-type: none">- Submit all code in a GitHub repository.- Include scripts for data loading, training, and evaluation.- GitHub should be titled 'CS3-[FirstName-LastName]'• References:<ul style="list-style-type: none">- Include on a separate page in IEEE citation style.
Written Portion

- Problem Summary: Describe the task of kidney tumor segmentation and its relevance.
- Project Plan: Outline your modeling approach and include a process diagram.
- Results and Interpretation: Discuss performance metrics and their meaning.
- Reflection: Share challenges, resolutions, and future improvements.

Code

- Data Preprocessing: Scripts for loading and preparing dataset.
- Model Training: PyTorch code for training with train-test split.
- Evaluation: Dice score, Jaccard index, precision, and recall.
- Documentation: Well-commented code and a README with instructions.

References

- Include a bibliography in IEEE style.
- Cite external sources used, excluding those already provided.